Foundation of Computer Science 1

©Laura Kovács

Revision 2
November 18, 2009

Problem 1. Consider the binary relation $\approx_3$ over the integer numbers $\mathbb{Z}$ defined as:

$$a \approx_3 b \quad \text{if and only if } a - b \text{ is a multiple of 3} \qquad \text{(where } a, b \in \mathbb{Z}\text{)}$$

Prove that $\approx_3$ is an equivalence relation!

Solution. For proving that $\approx_3$ is an equivalence relation, we need to prove that:

1. $\approx_3$ is reflexive;
2. $\approx_3$ is symmetric;
3. $\approx_3$ is transitive.

1. Proving that $\approx_3$ is reflexive.

$$\approx_3 \text{ is reflexive iff } \forall a : a \in \mathbb{Z} : a \approx_3 a.$$

We hence prove that $\forall a : a \in \mathbb{Z} : a \approx_3 a$.

We take *arbitrary* $a \in \mathbb{Z}$, and *prove* $a \approx_3 a$.
That is, we prove that $a - a$ is a multiple of 3.

Since $a - a = 0$ and 0 is a multiple of 3, we have that $a - a$ is a multiple of 3.
Thus $a \approx_3 a$ for arbitrary $a \in \mathbb{Z}$.

Hence $\approx_3$ is reflexive.

2. Proving that $\approx_3$ is symmetric.

$$\approx_3 \text{ is symmetric iff } \forall a : a, b \in \mathbb{Z} : a \approx_3 b \implies b \approx_3 a.$$

We hence prove that $\forall a, b : a, b \in \mathbb{Z} : a \approx_3 b \implies b \approx_3 a$.

We take *arbitrary* $a, b \in \mathbb{Z}$ and *assume* $a \approx_3 b$.
We *prove* that $b \approx_3 a$.

Since $a \approx_3 b$, we know that $a - b$ is a multiple of 3.
We can thus write that $a - b = 3 * k$, for some $k \in \mathbb{Z}$.

Then, $b - a = -(a - b) = -(3 * k) = 3 * (-k)$, where $-k \in \mathbb{Z}$.
Hence, $b - a$ is a multiple of 3, and thus $b \approx_3 a$.

This concludes that $\approx_3$ is symmetric.

3. Proving that $\approx_3$ is transitive.

$$\approx_3 \text{ is transitive iff } \forall a, b, c : a, b, c \in \mathbb{Z} : a \approx_3 b \wedge b \approx_3 c \implies a \approx_3 c.$$

We hence prove that $\forall a, b, c : a, b, c \in \mathbb{Z} : a \approx_3 b \ \wedge \ b \approx_3 c \implies a \approx_3 c$.

We take *arbitrary* $a, b, c \in \mathbb{Z}$ and *assume* $a \approx_3 b$ and $b \approx_3 c$.
We *prove* that $a \approx_3 c$.

Since $a \approx_3 b$, we know that $a - b$ is a multiple of 3.
We can thus write:

$$a - b = 3 * k_1, \text{ for some } k_1 \in \mathbb{Z}. \tag{1}$$

Similarly, since $b \approx_3 c$, we know that $b - c$ is a multiple of 3.
We can thus write:

$$b - c = 3 * k_2, \text{ for some } k_2 \in \mathbb{Z}. \tag{2}$$

Then,

$$a - c \ = \ (a - b) + (b - c) \underset{(1),\,(2)}{=} 3 * k_1 + 3 * k_2 \ = \ 3 * (k_1 + k_2), \text{ where } k_1 + k_2 \in \mathbb{Z}.$$

Hence, $a - c$ is a multiple of 3, and thus $a \approx_3 c$.

This concludes that $\approx_3$ is transitive.

As $\approx_3$ is reflexive, symmetric and transitive, it is an equivalence relation.

Problem 2.

(2.1) Consider the binary relation $\gg$ over the natural numbers $\mathbb{N}$ defined as:

$$a \gg b \quad \text{if and only if } b = a^r \text{ for some } r \in \mathbb{N} \qquad \text{(where } a, b \in \mathbb{N})$$

(a) Prove that $\gg$ is a partial order!

Solution. For proving that $\gg$ is a partial order, we need to prove that:

(a.1) $\gg$ is reflexive;
(a.2) $\gg$ is antisymmetric;
(a.3) $\gg$ is transitive.

(a.1) Proving that $\gg$ is reflexive.

$$\gg \text{ is reflexive iff } \forall a : a \in \mathbb{N} : a \gg a.$$

We hence prove that $\forall a : a \in \mathbb{N} : a \gg a$.

We take *arbitrary* $a \in \mathbb{N}$, and *prove* $a \gg a$.
That is, we prove that $a = a^r$ for some $r \in \mathbb{N}$.

Since $a = a^1$, we have that $a = a^r$ for $r = 1$.
Thus $a \gg a$ for arbitrary $a \in \mathbb{Z}$.

Hence $\gg$ is reflexive.

(a.2) Proving that $\gg$ is antisymmetric.

$$\gg \text{ is antisymmetric iff } \forall a : a, b \in \mathbb{N} : a \gg b \ \wedge \ b \gg a \implies a = b.$$

We hence prove that $\forall a, b : a, b \in \mathbb{N} : a \gg b \ \wedge \ b \gg a \implies a = b$.

We take *arbitrary* $a, b \in \mathbb{N}$ and *assume* $a \gg b$ and $b \gg a$.
We *prove* that $a = b$.

Since $a \gg b$, we know that $b = a^{r_1}$ for some $r_1 \in \mathbb{N}$.

Similarly, since $b \gg a$, we know that $a = b^{r_2}$ for some $r_2 \in \mathbb{N}$.

Then, $a = b^{r_2} = (a^{r_1})^{r_2} = a^{r_1 * r_2}$, and hence $r_1 * r_2 = 1$.
Since $r_1, r_2 \in \mathbb{N}$, from $r_1 * r_2 = 1$ we conclude that $r_1 = r_2 = 1$.
That is, $a = b^{r_2} = b$. (Similarly, $b = a^{r_1} = a$).
Thus, $a = b$.

This concludes that $\gg$ is antisymmetric.

(a.3) Proving that $\gg$ is transitive.

$\gg$ is transitive iff $\forall a, b, c : a, b, c \in \mathbb{N} : a \gg b \ \wedge \ b \gg c \implies a \gg c$.

We hence prove that $\forall a, b, c : a, b, c \in \mathbb{N} : a \gg b \ \wedge \ b \gg c \implies a \gg c$.

We take *arbitrary* $a, b, c \in \mathbb{N}$ and *assume* $a \gg b$ and $b \gg c$.
We *prove* that $a \gg c$.

Since $a \gg b$, we know that $b = a^{r_1}$ for some $r_1 \in \mathbb{N}$.

Similarly, since $b \gg c$, we know that $c = b^{r_2}$ for some $r_2 \in \mathbb{N}$.

Then,

$$c = b^{r_2} = (a^{r_1})^{r_2} = a^{r_1 * r_2}, \text{ where } r_1 * r_2 \in \mathbb{N}.$$

Hence, $a \gg c$.

This concludes that $\gg$ is transitive.

As $\gg$ is reflexive, antisymmetric and transitive, it is a partial order.

(b) Consider the set $A = \{1, 2, 4, 16\} \subset \mathbb{N}$. Give three different upper bounds of A with respect to the relation $\gg$. What is $lub(A)$?

Solution.

$y \in \mathbb{N}$ is an upper bound of A iff $\forall x : x \in A : x \gg y$ is valid.
That is, $y \in \mathbb{N}$ is an upper bound of A iff:

$$(1 \gg y) \ \wedge \ (2 \gg y) \ \wedge \ (4 \gg y) \ \wedge (16 \gg y) \quad \text{is} \quad True$$

Namely, $y \in \mathbb{N}$ is an upper bound of A iff:

- $1 \gg y$ is $True$, that is $y = 1^{r_1}$ for some $r_1 \in \mathbb{N}$;
- $2 \gg y$ is $True$, that is $y = 2^{r_2}$ for some $r_2 \in \mathbb{N}$;
- $4 \gg y$ is $True$, that is $y = 4^{r_3}$ for some $r_3 \in \mathbb{N}$;
- $16 \gg y$ is $True$, that is $y = 16^{r_4}$ for some $r_4 \in \mathbb{N}$.

Note that for any upper bound $y \in \mathbb{N}$ of A we have that $y = 1^{r_1}$ for some $r_1 \in \mathbb{N}$.
Therefore, the only *possible* upper bound of A is $y = 1$.

Further,
- As $1 = 2^0$, we conclude that $2 \gg 1$
- As $1 = 4^0$, we conclude that $4 \gg 1$
- As $1 = 16^0$, we conclude that $16 \gg 1$.

Hence,

$$(1 \gg 1) \ \wedge \ (2 \gg 1) \ \wedge \ (4 \gg 1) \ \wedge \ (16 \gg 1) \quad \text{is} \quad True,$$

and $y = 1$ is an upper bound of A.
As mentioned before, there are no other upper bounds of A.

Since $y = 1$ is the only upper bound of A, it is also the $lub(A)$.
That is, $lub(A) = 1$.

Remark: If A would have not contained 1, then there would have been infinitely many upper bounds of A.
Namely, consider now $A = \{2, 4, 16\}$. Then, for example,
- 1 is an upper bound of A;
- 16 is an upper bound of A;
- 256 is an upper bound of A;
- and essentially any power of 16 is an upper bound of A. That is 16^r is an upper bound for A, for any $r \in \mathbb{N}$.

Finally, the $lub(\{2, 4, 16\}) = 16$.

(c) Is $\gg$ an equivalence relation? Justify your answer!

Solution.

No, $\gg$ is NOT an equivalence relation!

Justification.
$\gg$ is an equivalence relation iff it is reflexive, symmetric and transitive.
We have already proved that $\gg$ is reflexive and transitive.
However, $\gg$ is NOT symmetric (see below). Therefore, $\gg$ is not an equivalence relation.

$\gg$ is symmetric iff the formula $\forall a, b : a, b \in \mathbb{N} : a \gg b \implies b \gg a$ is valid.
However, $\forall a, b : a, b \in \mathbb{N} : a \gg b \implies b \gg a$ is NOT a valid formula.
Take *for example* $a = 2$ and $b = 4$, for which $a \gg b$ holds as $4 = 2^2$.
However, $2 \neq 4^r$ for any $r \in \mathbb{N}$, and therefore $b \gg a$ does NOT hold.
That is, for $a = 2$ and $b = 4$ we have that $a \gg b$ is $True$, but $b \gg a$ is $False$.
Hence, $\gg$ is not symmetric.

d) Is $\gg$ a total order? Justify your answer!

Solution.

No, $\gg$ is NOT a total order!

Justification.
$\gg$ is a total order iff it is a partial order and it is a total relation.
We have already proved that $\gg$ is a partial order.

However, $\gg$ is NOT a total relation (see below). Therefore, $\gg$ is not a total order.

$\gg$ is a total relation iff the formula $\forall a, b : a, b \in \mathbb{N} : a \gg b \ \vee \ b \gg a$ is valid.
However, $\forall a, b : a, b \in \mathbb{N} : a \gg b \ \vee \ b \gg a$ is NOT a valid formula.
Take *for example* $a = 2$ and $b = 3$.
Then, $a \gg b$ is $False$, as $3 \neq 2^r$ for any $r \in \mathbb{N}$.
Similarly, $b \gg a$ is $False$, as $2 \neq 3^r$ for any $r \in \mathbb{N}$.
That is, for $a = 2$ and $b = 3$ we have that $a \gg b \ \vee \ b \gg a$ is $False$.
Hence, $\gg$ is not a total relation.

Problem 3. Let s_1, s_2 and s_3 be program statements, and consider Q be a predicate formula over program variables. What are the truth values of the following statements?

(3.1) $\mathrm{wp}(s_1; s_2; s_3, Q) = \mathrm{wp}(s_1; s_2, \mathrm{wp}(s_3, Q))$;

Solution.

The given formula is $True$.

Justification.

$$\mathrm{wp}(s_1; s_2; s_3,\ Q) \underset{=}{\mathit{SequenceRule}} \mathrm{wp}(s_1; s_2,\ \mathrm{wp}(s_3, Q))$$

(3.2) $\mathrm{wp}(s_1; s_2; s_3, Q) = \mathrm{wp}(s_2; s_1, \mathrm{wp}(s_3, Q))$;

Solution.

The given formula is $False$.

Justification.

$$\mathrm{wp}(s_1; s_2; s_3,\ Q) \underset{=}{\mathit{SequenceRule}} \mathrm{wp}(s_1; s_2,\ \mathrm{wp}(s_3, Q)) \underset{\neq}{\mathit{in\ general}} \mathrm{wp}(s_2; s_1,\ \mathrm{wp}(s_3, Q))$$

(3.2) $\mathrm{wp}(\underline{\text{while}}\ (True)\ \underline{\text{do}}\ s_1, Q) \implies \mathrm{wp}(s_1, \mathrm{wp}(\underline{\text{while}}\ (True)\ \underline{\text{do}}\ s_1, Q))$;

Solution.

The given formula is $True$.

Justification.

By definition, $\mathrm{wp}(\underline{\text{while}}\ (True)\ \underline{\text{do}}\ s_1, Q) =$ loop invariant.
Let us denote this loop invariant by I.
Then $\mathrm{wp}(\underline{\text{while}}\ (True)\ \underline{\text{do}}\ s_1, Q) = I$.
Since I is the loop invariant, we know that

(1) $I \wedge \underbrace{True}_{\text{loop condition}} \implies \mathrm{wp}(s_1, I)$ holds;

(2) $I \wedge \underbrace{False}_{\neg\text{loop condition}} \implies Q$ holds.

From the above listed property (1) of the invariant, we thus have:

$$
\begin{array}{c}
I \ \wedge \ True \implies \text{wp}(s_1, I) \\
\iff \\
I \implies \text{wp}(s_1, I).
\end{array}
$$

Hence, $I \implies \text{wp}(s_1, I)$ is $True$, where $I = \text{wp}(\underline{\text{while}}\ (True)\ \underline{\text{do}}\ s_1, Q)$.

Problem 4. Let x and y be program variables with values from the natural numbers $\mathbb{N}$.

(4.1) What is $\text{wp}(x := x + 1,\ x \leq 10)$?

Solution.

$$
\begin{array}{rl}
 & \text{wp}(\underline{x := x+1},\ \underline{x} \leq 10) = \\
\underset{=}{\scriptstyle AssignmentRule} & (x + 1 \leq 10) \\
= & x \leq 9
\end{array}
$$

(4.2) What is $\text{wp}(x := x + 1; y := y + x,\ x \leq 10)$?

Solution.

$$
\begin{array}{rl}
 & \text{wp}(x := x+1; y := y+x,\ x \leq 10) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(x := x+1,\ \text{wp}(\underline{y := y+x},\ x \leq 10)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{x := x+1},\ \underline{x} \leq 10) \\
\underset{=}{\scriptstyle AssignmentRule} & x + 1 \leq 10 \\
= & x \leq 9
\end{array}
$$

(4.3) What is $\text{wp}(y := y + x; x := x + 1,\ x + y \leq 10)$?

Solution.

$$
\begin{array}{rl}
 & \text{wp}(y := y+x; x := x+1,\ x + y \leq 10) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(y := y+x,\ \text{wp}(\underline{x := x+1},\ \underline{x} + y \leq 10)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{y := y+x},\ x + 1 + \underline{y} \leq 10) \\
\underset{=}{\scriptstyle AssignmentRule} & x + y + x \leq 9 \\
= & 2 * x + y \leq 9
\end{array}
$$

(4.4) What is $\text{wp}(x := x + 1; y := y + x,\ x + y \leq 10)$?

Solution.

$$
\begin{array}{rl}
 & \text{wp}(x := x+1; y := y+x,\ x + y \leq 10) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(x := x+1,\ \text{wp}(\underline{y := y+x},\ x + \underline{y} \leq 10)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{x := x+1},\ \underline{x} + y + \underline{x} \leq 10) \\
\underset{=}{\scriptstyle AssignmentRule} & x + 1 + y + x + 1 \leq 10 \\
= & 2 * x + y \leq 8
\end{array}
$$

(4.5) What is wp($x := x+1; y := y+x,\ True$)?

Solution.

$$\begin{array}{rl}
& \text{wp}(x := x+1; y := y+x,\ True) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(x := x+1,\ \text{wp}(\underline{y := y+x},\ True)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{x := x+1},\ True) \\
\underset{=}{\scriptstyle AssignmentRule} & True
\end{array}$$

(4.6) What is wp($x := x+1; x := x-1,\ x+y \leq 10$)?

Solution.

$$\begin{array}{rl}
& \text{wp}(x := x+1; x := x-1,\ x+y \leq 10) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(x := x+1,\ \text{wp}(\underline{x := x-1},\ \underline{x}+y \leq 10)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{x := x+1},\ \underline{x}-1+y \leq 10) \\
\underset{=}{\scriptstyle AssignmentRule} & x+1-1+y \leq 10 \\
= & x+y \leq 10
\end{array}$$

(4.7) What is wp($y := x-1; x := y+1,\ x+y \leq 10$)?

Solution.

$$\begin{array}{rl}
& \text{wp}(y := x-1; x := y+1,\ x+y \leq 10) = \\
\underset{=}{\scriptstyle SequenceRule} & \text{wp}(y := x-1,\ \text{wp}(\underline{x := y+1},\ \underline{x}+y \leq 10)) \\
\underset{=}{\scriptstyle AssignmentRule} & \text{wp}(\underline{y := x-1},\ \underline{y}+1+\underline{y} \leq 10) \\
\underset{=}{\scriptstyle AssignmentRule} & x-1+1+x-1 \leq 10 \\
= & 2x \leq 11
\end{array}$$

(4.8) What is wp($\underline{\text{if}}\ (x>5)\ \underline{\text{then}}\ x := x-1\ \underline{\text{else}}\ x := x+1,\ x+y \leq 10$)?

Solution.

$$\begin{array}{rl}
& \text{wp}(\underline{\text{if}}\ (x>5)\ \underline{\text{then}}\ x := x-1\ \underline{\text{else}}\ x := x+1,\ x+y \leq 10) = \\
\underset{=}{\scriptstyle ConditionalRule} & \big(x > 5 \implies \text{wp}(\underline{x := x-1},\ \underline{x}+y \leq 10)\big) \\
& \wedge \\
& \big(x \leq 5 \implies \text{wp}(\underline{x := x+1},\ \underline{x}+y \leq 10)\big) \\
\underset{=}{\scriptstyle AssignmentRule} & \big(x > 5 \implies x+y \leq 11\big) \ \wedge \big(x \leq 5 \implies x+y \leq 9\big)
\end{array}$$

(4.9) What is wp($\underline{\text{if}}\ (x>5)\ \underline{\text{then}}\ x := x-1; y := y-x\ \underline{\text{else}}\ x := x+1; y := y+x,\ x+y \leq 10$)?

Solution.

$$\text{wp}(\underline{\text{if}}\ (x>5)\ \underline{\text{then}}\ x:=x-1; y:=y-x\ \underline{\text{else}}\ x:=x+1; y:=y+x,\ x+y\leq 10) =$$

$$\overset{ConditionalRule}{=} \big(\ x>5 \implies \text{wp}(x:=x-1; y:=y-x,\ x+y\leq 10)\ \big)$$
$$\wedge$$
$$\big(\ x\leq 5 \implies \text{wp}(x:=x+1; y:=y+x,\ x+y\leq 10)\ \big)$$

$$\overset{SequenceRule}{=} \big(\ x>5 \implies \text{wp}(x:=x-1,\ \text{wp}(\underline{y:=y-x},\ x+\underline{y}\leq 10))\ \big)$$
$$\wedge$$
$$\big(\ x\leq 5 \implies \text{wp}(x:=x+1,\ \text{wp}(\underline{y:=y+x},\ x+\underline{y}\leq 10))\ \big)$$

$$\overset{AssignmentRule}{=} \big(x>5 \implies \text{wp}(x:=x-1,\ x+y-x\leq 10)\big)$$
$$\wedge$$
$$\big(x\leq 5 \implies \text{wp}(x:=x+1,\ \ x+y+x\leq 10)\big)$$

$$= \big(x>5 \implies \text{wp}(\underline{x:=x-1},\ y\leq 10)\big)$$
$$\wedge$$
$$\big(x\leq 5 \implies \text{wp}(\underline{x:=x+1},\ \ 2*\underline{x}+y\leq 10)\big)$$

$$\overset{AssignmentRule}{=} \big(x>5 \implies y\leq 10\big)\ \wedge \big(x\leq 5 \implies 2*x+y\leq 8\big)$$

Problem 5. Let x and y be program variables with values from the integer numbers $\mathbb{Z}$. Consider the Hoare triple:

$$\{x=1\wedge y=1\}\quad \underline{\text{while}}\ (x<10)\ \underline{\text{do}}\ x:=x+1; y:=y+1\ \underline{\text{end while}}\quad \{x=10\wedge y=10\},$$

annotated with the loop invariant $(x\leq 10\wedge x=y)$.

What are the verification conditions of the above given Hoare triple?

Solution.

There are 3 verification conditions, listed below in blue.

Note: The simplified verification conditions given in red were not required to be computed in this exercise! It is listed here so one can see how establishing correctness (i.e. True truth value) of verification conditions for loop verification can be done.

Verification Condition 1.

$$\underbrace{x=1\wedge y=1}_{\text{precondition}} \implies \underbrace{x\leq 10\wedge x=y}_{\text{invariant}}$$

$$\iff \mathit{True}$$

Assume $x=1$ and $y=1$. *Prove* $x\leq 10$ and $x=y$.

As $x=1$, then $x\leq 10$ holds (as $1\leq 10$ is True).

As $x=1$ and $y=1$, then $x=y$ also holds (as $1=1$ is True).

Verification Condition 2.

$$\underbrace{x \le 10 \wedge x = y}_{\text{invariant}} \quad \wedge \quad \underbrace{(x < 10)}_{\text{loop condition}} \quad \Longrightarrow \text{wp}(\underbrace{x := x+1; y := y+1}_{\text{loop body}}, \underbrace{x \le 10 \wedge x = y}_{\text{invariant}})$$

$$\overset{\textit{SequenceRule}}{\Longleftrightarrow} \quad x \le 10 \wedge x = y \wedge x < 10 \Longrightarrow \text{wp}(x := x+1,\ \text{wp}(\underline{y := y+1},\ x \le 10 \wedge x = \underline{y}))$$

$$\overset{\textit{AssignmentRule}}{\Longleftrightarrow} \quad x \le 10 \wedge x = y \wedge x < 10 \Longrightarrow \text{wp}(\underline{x := x+1},\ \underline{x} \le 10 \wedge \underline{x} = y+1)$$

$$\Longleftrightarrow \quad x \le 10 \wedge x = y \wedge x < 10 \Longrightarrow x+1 \le 10 \wedge x+1 = y+1$$

$$\Longleftrightarrow \quad x \le 10 \wedge x = y \wedge x < 10 \Longrightarrow x \le 9 \wedge x = y$$

$\Longleftrightarrow$ $\quad True$

Assume $x \le 10$ and $x = y$ and $x < 10$. *Prove* $x \le 9$ and $x = y$.

As $x \le 10$ and $x < 10$, we know $x < 10$, that is $x \le 9$. Hence, $x \le 9$ holds.

As $x = y$, then $x = y$ obviously holds.

Verification Condition 3.

$$\underbrace{x \le 10 \wedge x = y}_{\text{invariant}} \quad \wedge \quad \underbrace{x \ge 10}_{\neg\text{loop condition}} \quad \Longrightarrow \quad \underbrace{x = 10 \wedge y = 10}_{\text{postcondition}}$$

$\Longleftrightarrow$ $\quad True$

Assume $x \le 10$ and $x = y$ and $x \ge 10$. *Prove* $x = 10$ and $y = 10$.

As $x \le 10$ and $x \ge 10$, we know $x = 10$ (antisymmetry of $\le$). Thus, $x = 10$ holds.

As $x = y$, then, since $x = 10$, we have $y = 10$. Thus, $y = 10$ also holds.

Problem 6. Let x and y be program variables with values from the natural numbers $\mathbb{N}$. Consider the Hoare triple:

$$\{x = 1\} \quad \underline{\text{while}}\ (x < 10)\ \underline{\text{do}}\ x := x + 1\ \underline{\text{end while}} \quad \{x = 10\}.$$

What are the truth values of the following statements?

(6.1) $x \le 10$ is an invariant;

Solution.

$x \le 10$ is an invariant.

Truth value of "$x = 10$ is an invariant" is $True$.

Justification.

$x \le 10$ is an invariant of the Hoare triple iff the verification conditions below are $True$.

1. $\underbrace{x = 1}_{\text{precondition}} \Longrightarrow \underbrace{x \le 10}_{\text{invariant}}$

$$\iff \textit{True}$$ *Assume $x = 1$. Prove $x \leq 10$.*

As $x = 1$, then $x \leq 10$ holds since $1 \leq 10$ is True.

2. $\underbrace{x \leq 10}_{\text{invariant}} \wedge \underbrace{x < 10}_{\text{loop condition}} \implies \mathrm{wp}(\underbrace{x := x+1}_{\text{loop body}}, \underbrace{x \leq 10}_{\text{invariant}})$

$$\underset{\textit{AssignmentRule}}{\iff} \quad x \leq 10 \wedge x < 10 \implies x + 1 \leq 10$$

$$\iff \quad x \leq 10 \wedge x < 10 \implies x \leq 9$$

$$\iff \quad \textit{True}$$ *Assume $x \leq 10$ and $x < 10$. Prove $x \leq 9$.*

As $x \leq 10$ and $x < 10$, we know $x < 10$, that is $x \leq 9$ holds.

3. $\underbrace{x \leq 10}_{\text{invariant}} \wedge \underbrace{x \geq 10}_{\neg\text{loop condition}} \implies \underbrace{x = 10}_{\text{postcondition}}$

$$\iff \textit{True}$$ *Assume $x \leq 10$ and $x \geq 10$. Prove $x = 10$.*

As $x \leq 10$ and $x \geq 10$, then $x = 10$ holds (by antisymmetry of $\leq$).

(6.2) $x < 10$ is an invariant;

Solution.

$x < 10$ is NOT an invariant.

Truth value of "$x < 10$ is an invariant" is *False*.

Justification.

$x < 10$ is an invariant of the Hoare triple iff the verification conditions below are *True*. But one of the verification conditions is False.

1. $\underbrace{x = 1}_{\text{precondition}} \implies \underbrace{x < 10}_{\text{invariant}}$

$$\iff \textit{True}$$ *Assume $x = 1$. Prove $x < 10$.*

As $x = 1$, then $x < 10$ holds since $1 < 10$ is *True*.

2. $\underbrace{x < 10}_{\text{invariant}} \wedge \underbrace{x < 10}_{\text{loop condition}} \implies \mathrm{wp}(\underbrace{x := x+1}_{\text{loop body}}, \underbrace{x < 10}_{\text{invariant}})$

$$\underset{\textit{AssignmentRule}}{\iff} \quad x < 10 \implies x + 1 < 10$$

$$\iff \quad x < 10 \implies x < 9$$

$\iff$ $False$ *Counterexample Proof*: Consider a *concrete value* of x such that $x < 10$.

Show that *for this value* of x, $x < 9$ does not hold.

Take x to be 9.

Then $x < 10$ is $True$, since $9 < 10$ is $True$.

But $x < 9$ is $False$, as $9 < 9$ is $False$.

3. $\underbrace{x < 10}_{\text{invariant}} \wedge \underbrace{x \geq 10}_{\neg\text{loop condition}} \implies \underbrace{x = 10}_{\text{postcondition}}$

$\iff$ $False \implies x = 10$ $x < 10$ and $x \geq 10$ is $False$

$\iff$ $True$

(6.3) $x = 10$ is an invariant.

Solution.

$x = 10$ is NOT an invariant.

Truth value of "$x = 10$ is an invariant" is $False$.

Justification.

$x = 10$ is an invariant of the Hoare triple iff the verification conditions below are $True$. But one of the verification conditions is False.

1. $\underbrace{x = 1}_{\text{precondition}} \implies \underbrace{x = 10}_{\text{invariant}}$

$\iff$ $False$ *Assume* $x = 1$. *Prove* $x = 10$.

As $x = 1$, then $1 = 10$ is $False$, and hence $x = 10$ does NOT hold.

2. $\underbrace{x = 10}_{\text{invariant}} \wedge \underbrace{x < 10}_{\text{loop condition}} \implies \text{wp}(\underbrace{x := x + 1}_{\text{loop body}}, \underbrace{x = 10}_{\text{invariant}})$

$\overset{AssignmentRule}{\iff}$ $False \implies x + 1 = 10$

$\iff$ $True$

3. $\underbrace{x = 10}_{\text{invariant}} \wedge \underbrace{x \geq 10}_{\neg\text{loop condition}} \implies \underbrace{x = 10}_{\text{postcondition}}$

$\iff$ $True$ *Assume* $x = 10$ and $x \geq 10$. *Prove* $x = 10$.

As $x = 10$ and $x \geq 10$, we know $x = 10$, and hence $x = 10$ holds.